

# Software-Projekt I

Prof. Dr. Rainer Koschke

Arbeitsgruppe Softwaretechnik  
Fachbereich Mathematik und Informatik  
Universität Bremen

Sommersemester 2013

# Anforderungsanalyse I

## 1 Anforderungsanalyse

- Lehrbücher
- Lernziele
- Herausforderungen
- Aktivitäten
- Ist-Analyse
  - Erhebungstechniken
  - Befragung
  - Beobachtung
- Soll-Analyse: Kreativitätstechniken
- Soll-Analyse: Prototyping
- Zusammenfassung der Techniken
- Anforderungsspezifikation
  - Bedeutung
  - Anzustrebende Eigenschaften
  - Regeln
  - Aufbau und Inhalt

# Das Problem der Anforderungsanalyse



Was der Kunde erklärte



Was der Projektleiter  
verstand



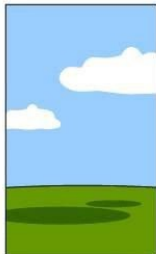
Wie es der Analytiker  
entwarf



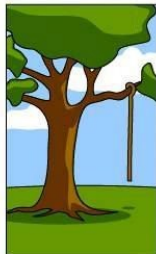
Was der Programmierer  
programmierte



Was der Berater definierte



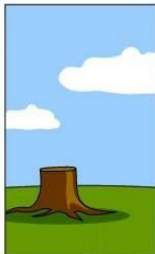
Wie das Projekt  
dokumentiert wurde



Was installiert wurde



Was dem Kunden in  
Rechnung gestellt wurde



Wie es gewartet wurde

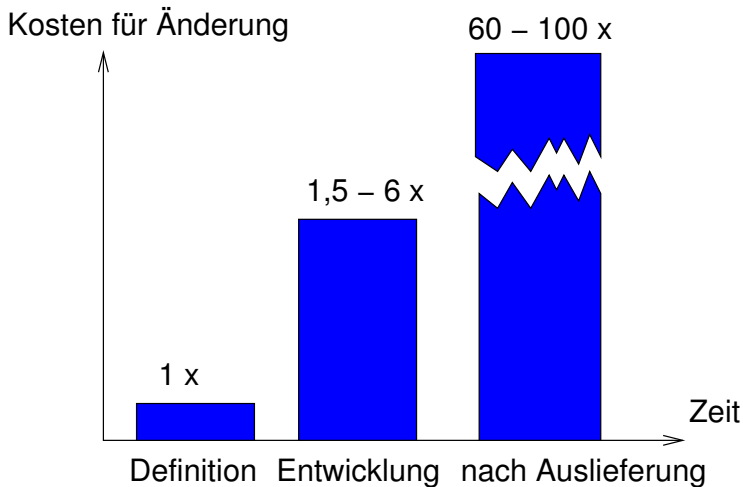


Was der Kunde wirklich  
gebraucht hätte



- Warum sind die Anforderungen so kritisch?
- Warum ist ihre Erhebung so schwer?

# Kosten für Änderungen



Pressman (2003)

# Warum die Anforderungsanalyse so schwer ist

- Kunden wissen häufig nicht, was sie genau wollen, bzw. können es nicht genau äußern
- Kunden sprechen ihre Sprache, die von Entwicklern nicht verstanden wird
- unterschiedliche Kundengruppen haben unterschiedliche Anforderungen, die sich mitunter widersprechen
- politische Entscheidungen können Anforderungen beeinflussen
- die Welt ändert sich, die Anforderungen an die Software auch; auch während der Entwicklung

– Sommerville (2004)

# Bewusstseinssebenen

- bewusstes Wissen (20-30%)
  - Wissen, über das man sich im Klaren ist oder das in seiner vollen Bedeutung klar erkannt wird

# Bewusstseinsebenen

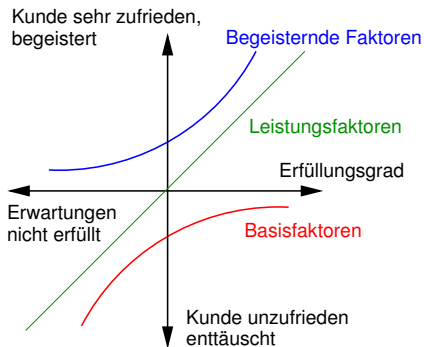
- bewusstes Wissen (20-30%)
  - Wissen, über das man sich im Klaren ist oder das in seiner vollen Bedeutung klar erkannt wird
- unbewusstes Wissen ( $\leq 40\%$ )
  - Wissen, das sich dem Bewusstsein im Moment nicht darbietet, aber dennoch handlungsbestimmend ist, und potenziell aufgerufen werden kann



# Bewusstseinssebenen

- bewusstes Wissen (20-30%)
  - Wissen, über das man sich im Klaren ist oder das in seiner vollen Bedeutung klar erkannt wird
- unbewusstes Wissen ( $\leq 40\%$ )
  - Wissen, das sich dem Bewusstsein im Moment nicht darbietet, aber dennoch handlungsbestimmend ist, und potenziell aufgerufen werden kann
- unterbewusstes Wissen
  - unbekannte Wünsche, die erst von außen herangetragen werden müssen, um als Anforderungen erkannt zu werden

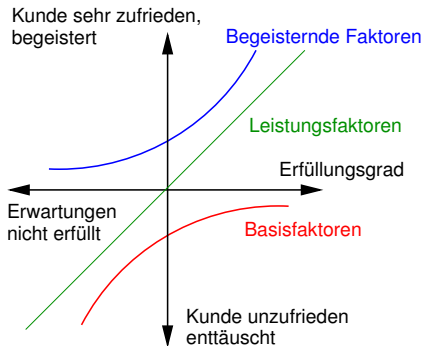
# Kano-Modell



## Basisfaktoren

- Minimalanforderungen
- Mangel führt zu massiver Unzufriedenheit
- mehr als Zufriedenheit ist nicht möglich

# Kano-Modell



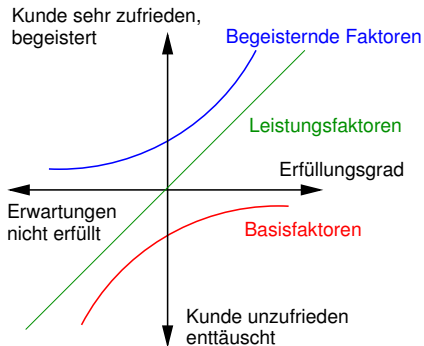
## Basisfaktoren

- Minimalanforderungen
- Mangel führt zu massiver Unzufriedenheit
- mehr als Zufriedenheit ist nicht möglich

## Leistungsfaktoren

- bewusst verlangte Sonderausstattung
- bei Erfüllung: Kundenzufriedenheit
- sonst: Unzufriedenheit

# Kano-Modell



## Basisfaktoren

- Minimalanforderungen
- Mangel führt zu massiver Unzufriedenheit
- mehr als Zufriedenheit ist nicht möglich

## Leistungsfaktoren

- bewusst verlangte Sonderausstattung
- bei Erfüllung: Kundenzufriedenheit
- sonst: Unzufriedenheit

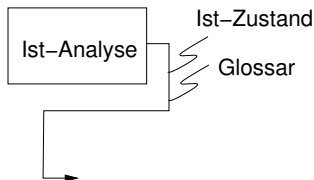
## Begeisternde Faktoren

- unbewusste Wünsche, nützliche/angenehme Überraschungen
- steigern Zufriedenheit überproportional



Was sind die wesentlichen Schritte zur Erhebung der Anforderungen?

# Schritte der Anforderungsanalyse: Ist-Analyse



**Ziel:** Verständnis der Welt, für die Softwarelösung angestrebt wird.

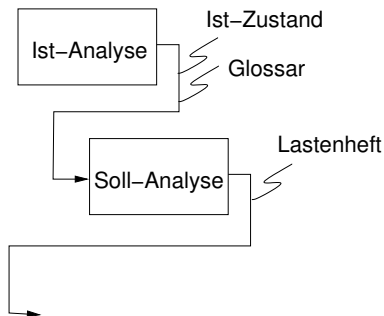
## Häufige Fehler:

- Entwickler sieht nicht, dass Kunde primär keine Veränderung, sondern Verbesserung anstrebt.
- Kunde beschreibt selten, was sich nicht ändern soll (weil es gut genug ist).
- Kunde  $\neq$  Endbenutzer; weiß nicht, was dieser braucht.

**Folgen von Mängeln:** Eigentliches Problem wird ignoriert.

**Erforderlich:** Beobachtungsgabe, Einfühlungsvermögen, Kommunikationsfähigkeit.

# Schritte der Anforderungsanalyse: Soll-Analyse



**Ziel:** Aufdeckung und Verbesserung bisheriger Schwächen durch Softwarelösung. Antizipation von Änderungen.

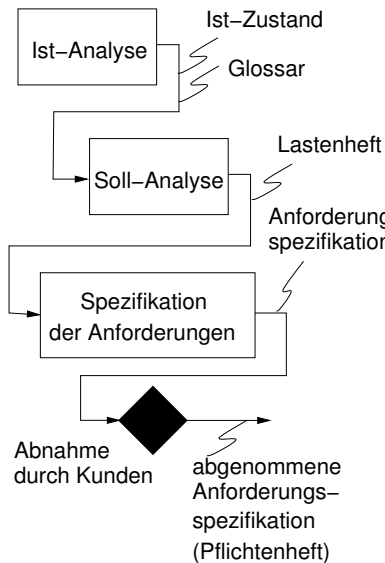
## Häufige Fehler:

- Entwickler gleiten in technische Details ab.
- Kunde hat keine klare Vorstellung bzw. kann sie nicht vermitteln.

**Folgen von Mängeln:** falsche Lösung wird spezifiziert.

**Erforderlich:** Analytische Fähigkeiten kombiniert mit Wissen über Machbarkeit von Softwarelösungen.

# Schritte der Anforderungsanalyse: Spezifikation



**Ziel:** Anforderungen genau beschreiben.

**Häufige Fehler:**

- Anforderungen bleiben vage
- Implementierungsdetails statt Anforderungen

**Folgen von Mängeln:**

- Vertragsstreitigkeiten am Projektende.

**Erforderlich:** Kommunikationsfähigkeit.



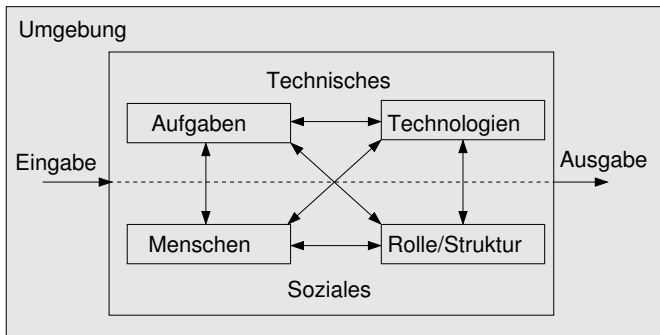


Was ist das Ziel und der Gegenstand der Ist-Analyse?

# Soziotechnisches System

## Definition

**Soziotechnisches System:** organisierte Menge von Menschen und Technologien, die in einer bestimmten Weise strukturiert sind, um eine Aufgabe zu erfüllen.



– Emery, Thorsrud & Trist 1964



# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen

... aus allen  
relevanten  
Blickwinkeln.

# Ist-Zustand: Was?

Verständnis von:

- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen

## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen

# Ist-Zustand: Was?

Verständnis von:

- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen

## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen



# Ist-Zustand: Was?

Verständnis von:

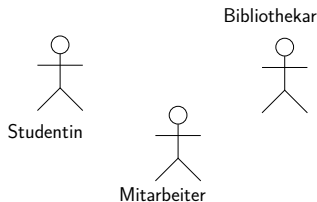
- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen



## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen



# Ist-Zustand: Was?

Verständnis von:

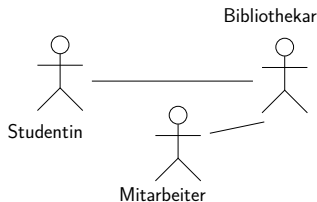
- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen



## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen





# Ist-Zustand: Was?

Verständnis von:

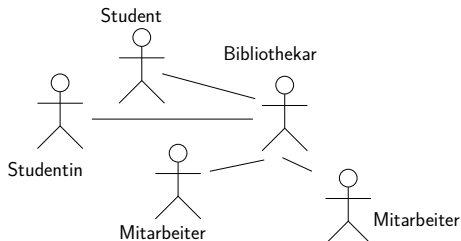
- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen



## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen



# Ist-Zustand: Was?

Verständnis von:

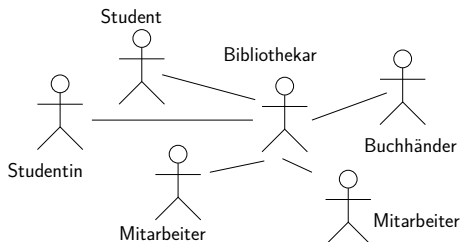
- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen



## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen



# Ist-Zustand: Was?

Verständnis von:

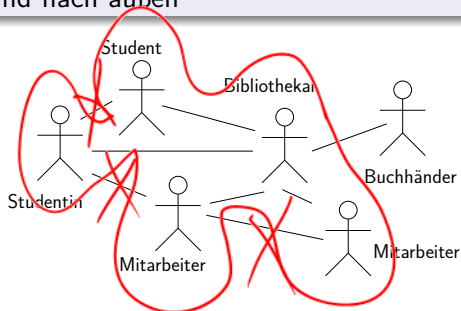
- **Struktur**
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen



## Bestandteile

organisatorisches Gefüge des Systems, für das Softwarelösung angestrebt wird

- relevante Akteure
- Systemgrenzen
- Art und Umfang der Verbindungen innerhalb und nach außen



## Ist-Zustand: relevante Akteure

Grundsatz: Kenne deinen Benutzer!

## Ist-Zustand: relevante Akteure

Grundsatz: Kenne deinen Benutzer!

**Aber:** *Der* Benutzer bzw. *die* Benutzerin ist eine Illusion. Es sind individuelle Menschen, um die es geht.

## Ist-Zustand: relevante Akteure

Grundsatz: Kenne deinen Benutzer!

**Aber:** *Der* Benutzer bzw. *die* Benutzerin ist eine Illusion. Es sind individuelle Menschen, um die es geht.

Andererseits: Wir können nicht jeden betrachten und müssen zusammenfassen.

# Ist-Zustand: relevante Akteure

## Persona

(in archetypischer Psychologie) die Maske oder Erscheinung, die man der Welt präsentiert.

(in der Softwareergonomie) erzählerische Beschreibung charakterischer Eigenschaften und Verhalten eines Benutzers oder Kunden, die spezifische Details nennt, statt Verallgemeinerungen.



# Persona-Poster

Name (fiktiv)	<i>Bernd Bib</i>	<i>Susi Studi</i>	<i>Michel Mit</i>
Bild (fiktiv)			
Rolle	Bibliothekar	Leiherin	Leiher
Beruf	Bibliothekar	Studentin	wiss. Mitarbeiter
Motto	Bücher sind mein Leben	Lernen ist meine Leidenschaft	Worte sagen mehr als Bilder
Ziele	effizient den Überblick behalten	ab und zu schnell ein Buch ausleihen	Spezialliteratur vertiefen

..... sowie Aufgaben, Ideen, Wünsche, Vorlieben, persönliche Details etc.



# Bedeutung von Personas

- archetypische Benutzerbeschreibungen
- typisch für Zielgruppen
- decken deren Anforderungen, Bedürfnisse und Ziele ab

– Astrid Beck, FHT Esslingen

# Bedeutung von Personas

- archetypische Benutzerbeschreibungen
- typisch für Zielgruppen
- decken deren Anforderungen, Bedürfnisse und Ziele ab
- stellvertretend für die (anonymen) realen Benutzer bei:
  - Anforderungserhebung
  - Interaktions-Design
  - Usability-Test
  - Strukturierung des Handbuchs
  - Akzeptanztest

– Astrid Beck, FHT Esslingen

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- **Aufgaben**
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen

## Bestandteile

Umfang und Art der anfallenden Aufgaben (Operationen) und Besonderheiten im Ablauf.

- Was wird gemacht?  
→ *Bücher verliehen*
- Wer oder was führt Operation aus?  
→ *Bibliothekar und Leiherr*
- Wann und wie häufig?  
→ *nur werktags: 1 Mal/Tag*

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- **Aufgaben**
- Kommunikation
- Dokumenten
- Daten
- Schwachstellen

## Bestandteile (Forts.)

- Zu welchem Zweck?  
→ *Lehre und Forschung*
- Nach welchen Regeln wirken Operationen zusammen?  
→ *nur Bibliothekar händigt Buch aus*
- Was benutzt/produziert Operation?  
→ *Zeit und Wissen des Bibliothekars, Karteikasten, Bibliothek/Buch für Ausleiher*

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- **Kommunikation**
- Dokumenten
- Daten
- Schwachstellen

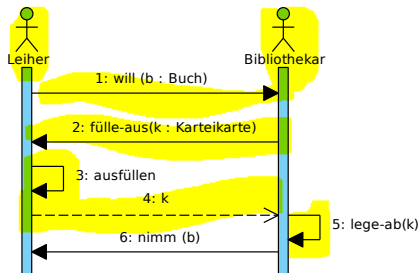
## Bestandteile

- Welche Vorrichtungen und Gelegenheiten zur Kommunikation gibt es (im Rahmen welcher Aufgaben)?  
→ *direkte Kommunikation im Büro, Telefon, E-Mail*
- **Wie läuft Kommunikation ab?**  
→ *initiiert vom Ausleiher*

# Ist-Zustand: Was?

Verständnis von:

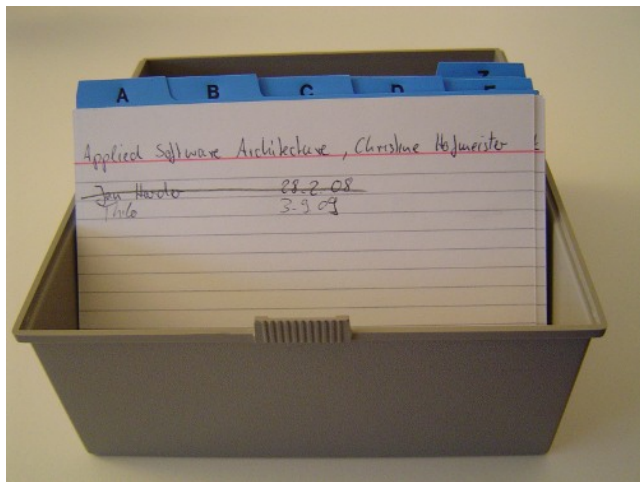
- Struktur
- Aufgaben
- **Kommunikation**
- Dokumenten
- Daten
- Schwachstellen



# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- **Dokumenten**
- Daten
- Schwachstellen



# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- **Dokumenten**
- Daten
- Schwachstellen

## Bestandteile

Dokumente, die verwendet und produziert werden

- Bezeichnung  
→ *Karteikarte*
- Inhalt  
→ *Buchtitel, erster Autor, Ausleiher, Datum der Ausleihe*
- Grad der Formalisierung, Aufbau  
→ *geringe Formalisierung (z.B. Nachname kann fehlen)*
- Verteiler  
→ *im Karteikasten, zugänglich für alle, die in die Bibliothek kommen*



# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- **Dokumenten**
- Daten
- Schwachstellen

## Bestandteile (Fortsetzung)

- Archivierung  
→ *keine*
- von wem produziert/verwendet?  
→ *produziert von Ausleiher/Bibliothekar*  
→ *verwendet von Bibliothekar für Rückgabe und Mahnung*

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- **Daten**
- Schwachstellen

## Bestandteile

Umfang und Art der verarbeiteten Daten

- Volumen  
→ *100 Bücher, 20 Ausleiher*
- Wachstum  
→ *10 weitere Bücher/Jahr*
- Wertebereiche  
→ *beliebig lange Namen und Titel,*  
→ *Ausleihdaten  $\geq 1.1.2008$ ,*  
→ *Anzahl Exemplare  $> 0$  etc.*
- Datenträger  
→ *Papier*

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- **Daten**
- Schwachstellen

## Bestandteile (Fortsetzung)

- **Ordnungsstrukturen**  
→ *Titel, Ausleihdatum*
- **Verarbeitungshäufigkeit**  
→ *ca. 1 Ausleihe/Tag*
- **Art und Erfordernisse der Datensicherung**  
→ *keine Datensicherung, wäre aber notwendig*
- **Abhängigkeiten zwischen den Daten**  
→ *es kann vom gleichen Buch mehrere Exemplare geben*

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- **Daten**
- Schwachstellen

## Bestandteile (Fortsetzung)

- Ordnungsstrukturen  
→ *Titel, Ausleihdatum*
- Verarbeitungshäufigkeit  
→ *ca. 1 Ausleihe/Tag*
- Art und Erfordernisse der Datensicherung  
→ *keine Datensicherung, wäre aber notwendig*
- Abhängigkeiten zwischen den Daten  
→ *es kann vom gleichen Buch mehrere Exemplare geben*

→ Resultat ist das Datenmodell (siehe später)

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- **Schwachstellen**

## Bestandteile

Untersuchung auf:

- Mängel
- Unvollständigkeiten
- Redundanzen

# Ist-Zustand: Was?

Verständnis von:

- Struktur
- Aufgaben
- Kommunikation
- Dokumenten
- Daten
- **Schwachstellen**

## Beispielschwachstellen

- Bibliothekar kann sich keinen leichten Überblick über die Ausleihe verschaffen
- Bibliothekar kann nicht ohne Weiteres mahnen
- Leiher kann nur in Bibliothek herausfinden, welche Bücher es gibt
- Neuanschaffungen werden nicht bekannt gegeben
- Leiher kann sich kein Buch vormerken
- nur anhand des Karteikastens in der Bibliothek ist zu bestimmen, ob ein Buch ausgeliehen ist
- wenn Karteikasten verloren geht, ist alle Information verloren



Welche Techniken können wir in der Ist-Analyse nutzen?

# Ist-Zustand: Wie?

## Erhebungstechniken

- Auswertung vorhandener Dokumente
- Befragung
  - schriftlicher Fragebogen
  - Interview
- Beobachtung
  - anekdotisch ↔ systematisch
  - teilnehmend ↔ nicht-teilnehmend
  - offen ↔ verdeckt
  - selbst ↔ fremd
  - Feld ↔ Labor



# Befragung

## Fragen zu Fragen:

- Wird die Frage verstanden?
- Bezugsrahmen der Befragten?
- Informationsstand der Befragten?
- Art der Frage?
- Anordnung der Fragen?
- Erhebungssituation (Interviewereinfluss)?
- Gründe für die Antwort der Befragten?

# Fragetypen: Geschlossene Fragen

Geschlossene Fragen:

*Welche Qualität hat die GUI? Bitte ankreuzen.*

☐ *sehr gut*

☐ *gut*

☐ *schlecht*

☐ *weiß nicht*

- Antwortalternativen vorgegeben
- auch Mehrfachantworten

# Fragetypen: Offene Fragen

Offene Fragen:

*Wie sollte die GUI verbessert werden?*

---

- Antworten in eigenen Worten, im eigenen Referenzsystem
- erfordert Ausdrucksfähigkeit der Befragten
- starker Einfluss des Fragenden, wenn präsent (durch Aufschreiben, Weglassen)
- hoher Auswertungsaufwand

# Fragetypen: Hybride Fragen

## Hybride Fragen:

*Was stört Sie an der GUI?*

- ☐ *lange Reaktionszeit*
- ☐ *mangelnde Selbsterklärungsfähigkeit*
- ☐ *fehlendes „Undo“*
- ☐ *umständliche Dialogführung*
- ☐ \_\_\_\_\_

- Kombination von geschlossenen und offenen Fragen

# Wann welche Erhebungsform?

## weniger strukturiertes Interview

- unstrukturiertes Untersuchungsgebiet
- offene Gesprächsführung und größere Antwortspielräume
- persönlicher Kontakt möglich
- Ortsbegehung möglich

## stark strukturierter Fragebogen

- vorstrukturiertes Untersuchungsgebiet
- gute Kenntnisse des Untersuchungsgebiets
- Operationalisierung der Hypothesen möglich

# Frageformulierung

- einfache Worte

# Frageformulierung

- einfache Worte

Stimmen Sie folgender Aussage zu?

„Die Repetition konsekutiver Transporte zu einer artifiziell konstruierten subterranean Dihydrogenmonoxidquelle impliziert das Potential zur Destabilisierung der physischen Struktur des utilisierten irdenen Gefäßes.“

# Frageformulierung

- einfache Worte

Stimmen Sie folgender Aussage zu?

„Die Repetition konsekutiver Transporte zu einer artifiziell konstruierten subterranean Dihydrogenmonoxidquelle impliziert das Potential zur Destabilisierung der physischen Struktur des utilisierten irdenen Gefäßes.“

Der Krug geht so lange zum Brunnen, bis er bricht.



# Frageformulierung

- einfache Worte
- kurz und konkret

„Wir möchten hier einmal in aller Deutlichkeit fragen, auch im Hinblick auf die noch offenen Punkte, die ihrerseits in der noch verbleibenden Zeit dieses Interviews mit Ihnen, für das wir uns hier im Übrigen an dieser Stelle noch einmal in aller Form bedanken möchten, nochmals aufgegriffen werden, ob und wenn ja, in welchem Grade Sie der Ansicht sind, dass diese Frage zu lang geraten ist.“

# Frageformulierung

- einfache Worte
- kurz und konkret
- keine doppelten Negationen
- neutral, nicht suggestiv

„Sie sind doch nicht etwa der Ansicht, dass die GUI nicht bedienbar ist, oder etwa doch?“

# Frageformulierung

- einfache Worte
- kurz und konkret
- keine doppelten Negationen
- neutral, nicht suggestiv
- nur auf einen Sachverhalt bezogen
- nicht hypothetisch

„Stellen Sie sich einmal vor, Sie wären im Pensionsalter und hätten eine erheblich eingeschränkte Sehschärfe, die auch die Farbenwahrnehmung beeinträchtigt. Würden Sie es unter diesen Umständen begrüßen, die Schriftgröße und die dargestellten Farben in der GUI einstellen zu können?“

# Frageformulierung

- einfache Worte
- kurz und konkret
- keine doppelten Negationen
- neutral, nicht suggestiv
- nur auf einen Sachverhalt bezogen
- nicht hypothetisch
- Befragten nicht überfordern
- balanciert (negative und positive Antwortmöglichkeiten)
- immer eine „weiß-nicht“-Kategorie bieten

# Beobachtung

## Prinzipien der Ethnographie<sup>1</sup>:

- Natürliche Umgebung
  - Aktivitäten in Alltagsumgebung untersuchen
- Ganzheitlichkeit
  - Einzelverhalten im Kontext untersuchen
- Beschreiben, nicht bewerten
  - Ist-Verhalten, nicht Soll-Verhalten
- Sicht der Handelnden einnehmen
  - Verhalten beschreiben in Begriffen, die für den Handelnden relevant und bedeutungsvoll sind

---

<sup>1</sup>teilnehmende Beobachtung in der Feldforschung

# Interview im Kontext

- ist eine Form der ethnographischen Untersuchung
- nach dem Meister-Lehrling-Modell
- Lernen durch Vormachen und Beobachten sowie Fragen und Klären
- geprägt durch
  - Bescheidenheit
  - Neugier
  - Aufmerksamkeit
  - konkrete (statt abstrakter) Fragen

# Beispiel eines Ablaufes

## ① Einleitung (15 min)

- ▶ Vorstellung, Ziele, Dank
- ▶ Zustimmung zu Aufzeichnung, Vertraulichkeit
- ▶ Arbeit, nicht Person wird betrachtet!
- ▶ Meinungen zu technischer Unterstützung?
- ▶ Überblick gewinnen

# Beispiel eines Ablaufes

## ① Einleitung (15 min)

- ▶ Vorstellung, Ziele, Dank
- ▶ Zustimmung zu Aufzeichnung, Vertraulichkeit
- ▶ Arbeit, nicht Person wird betrachtet!
- ▶ Meinungen zu technischer Unterstützung?
- ▶ Überblick gewinnen

## ② Übergang (1 min)

- ▶ Regeln, Rollen, Beziehung
- ▶ ich frage, Sie dürfen abwehren



# Beispiel eines Ablaufes

## ① Einleitung (15 min)

- ▶ Vorstellung, Ziele, Dank
- ▶ Zustimmung zu Aufzeichnung, Vertraulichkeit
- ▶ Arbeit, nicht Person wird betrachtet!
- ▶ Meinungen zu technischer Unterstützung?
- ▶ Überblick gewinnen

## ② Übergang (1 min)

- ▶ Regeln, Rollen, Beziehung
- ▶ ich frage, Sie dürfen abwehren

## ③ Erhebung im Kontext (2 Std.)

- ▶ Beobachtung und Nachfragen
- ▶ Notizen machen, mitlaufen, sich unsichtbar machen
- ▶ Pausen nach Wunsch

# Beispiel eines Ablaufes

## ① Einleitung (15 min)

- ▶ Vorstellung, Ziele, Dank
- ▶ Zustimmung zu Aufzeichnung, Vertraulichkeit
- ▶ Arbeit, nicht Person wird betrachtet!
- ▶ Meinungen zu technischer Unterstützung?
- ▶ Überblick gewinnen

## ② Übergang (1 min)

- ▶ Regeln, Rollen, Beziehung
- ▶ ich frage, Sie dürfen abwehren

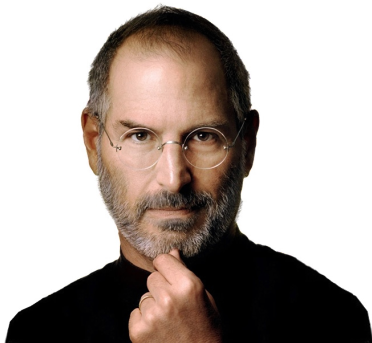
## ③ Erhebung im Kontext (2 Std.)

- ▶ Beobachtung und Nachfragen
- ▶ Notizen machen, mitlaufen, sich unsichtbar machen
- ▶ Pausen nach Wunsch

## ④ Zusammenfassung (15 min)

- ▶ was die Beschäftigte tut, ihre Rolle
- ▶ was wichtig ist
- ▶ Ergänzungen, Korrekturen?

*Users don't know what they  
want until we tell them.*



# Fragen



Wie können wir gute Ideen entwickeln?

# Fragen



Wie können wir dem Benutzer frühzeitig ein konkretes Bild vermitteln, wie wir das Problem zu lösen gedenken?

# Prototyping

## Zielsetzung:

- Anforderungen anhand eines Beispiels erheben und überprüfen
- technische Möglichkeiten überprüfen und demonstrieren
- frühzeitig mögliche Lösungsansätze präsentieren

## Idee:

- rasche und billige Entwicklung eines prototypischen Systems als Diskussionsgrundlage

# Prototyping

## Zielsetzung:

- Anforderungen anhand eines Beispiels erheben und überprüfen
- technische Möglichkeiten überprüfen und demonstrieren
- frühzeitig mögliche Lösungsansätze präsentieren

## Idee:

- rasche und billige Entwicklung eines prototypischen Systems als Diskussionsgrundlage

## Typen unterscheiden sich in ...

- Lebensdauer: Wegwerfprototyp, evolutionärer Prototyp
- Zweck: technische Machbarkeit, Demonstration der Funktionalität oder Interaktion

# Typen von Prototypen in Bezug auf Lebensdauer

## Wegwerf-Prototyp:

- beschreibt ein Softwaresystem exemplarisch
- dient zur Erhebung und Analyse von Anforderungen oder zur Überprüfung technischer Machbarkeit
- demonstriert die Funktionalität, die mit Stakeholdern diskutiert werden soll
- implementiert nicht notwendigerweise die gezeigte Funktionalität (z.B. GUI-Prototyp)
- ist als Komponente für das Endprodukt ungeeignet, weil billig erstellt





# Typen von Prototypen in Bezug auf Lebensdauer

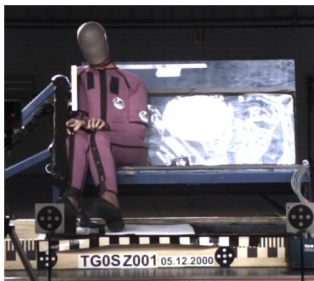
## Evolutionärer Prototyp:

- dient zur schnellen Bereitstellung eines funktionsfähigen Systems im Rahmen von evolutionären Prozessmodellen zur Softwareentwicklung
- wird in weiteren Ausbaustufen zum endgültigen Produkt weiterentwickelt

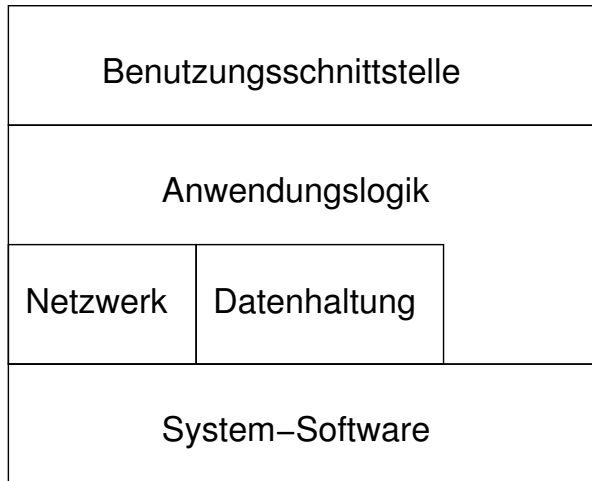
# Typen von Prototypen in Bezug auf Zweck

## Technischer Prototyp:

- zeigt die technische Umsetzbarkeit von Ansätzen zur Problemlösung
- implementiert einen (kleinen) Ausschnitt der Funktionalität des Systems
- wird eher zur Machbarkeitsabschätzung und -demonstration eingesetzt

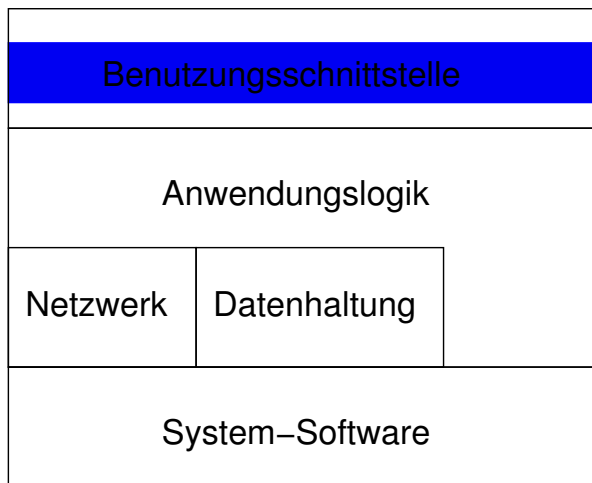


## Horizontaler vs. vertikaler technischer Prototyp



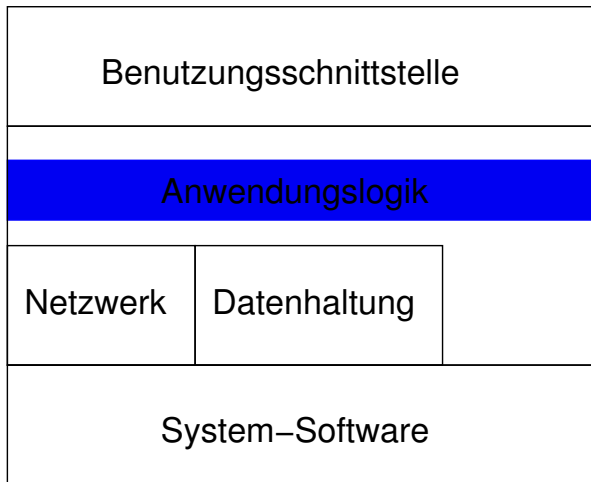
# Horizontaler vs. vertikaler technischer Prototyp

Horizontaler Prototyp: realisiert Aspekte einer spezifischen Ebene des Softwaresystems; Bsp: Oberflächenprototyp



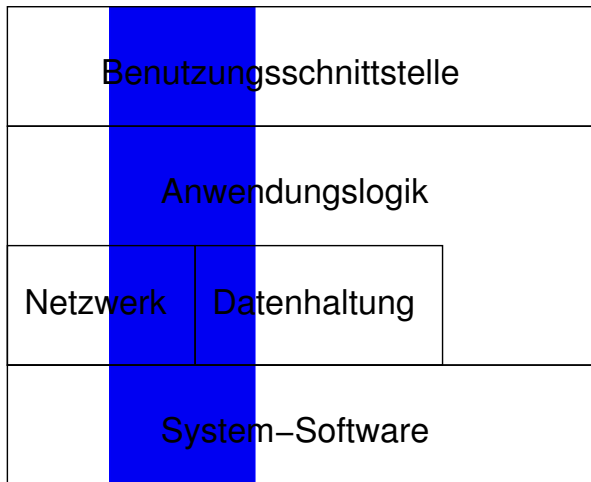
## Horizontaler vs. vertikaler technischer Prototyp

Horizontaler Prototyp: realisiert Aspekte einer spezifischen Ebene des Softwaresystems; Bsp: Anwendungsprototyp



# Horizontaler vs. vertikaler technischer Prototyp

Vertikaler Prototyp: realisiert ausgewählte Aspekte des Softwaresystems vollständig



# Prototypen für Interaktion: Storyboards

Storyboards:

- Prototypen für die Interaktion zwischen Mensch und Maschine
- demonstrieren das zu diskutierende Systemverhalten als „Geschichte“

Typen:

- passives Storyboard (Papierprototyp)
- aktives Storyboard (animierter Prototyp)
- interaktives Storyboard (ausführbarer Prototyp)

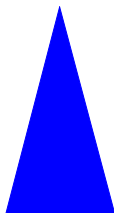


Abbildung:  
Aufwand

## Passive Storyboards

### **Demonstration:**

Analytiker spielt die Bedienung mit dem System durch, indem er entlang eines Anwendungsszenarios Eingabemöglichkeiten und Systemreaktionen demonstriert

### **Mittel:**

- Skizzen
- Bildschirm-Masken (Screenshots)
- mögliche Systemausgaben

### **Bemerkung:**

- + ermöglicht einfache und billige Prototyperstellung
- + ermöglicht Interaktion mit Beteiligten am Beispiel
- erfordert Anwesenheit des Analytikers



# Beispiel passives Storyboard



## Beispiel passives Storyboard

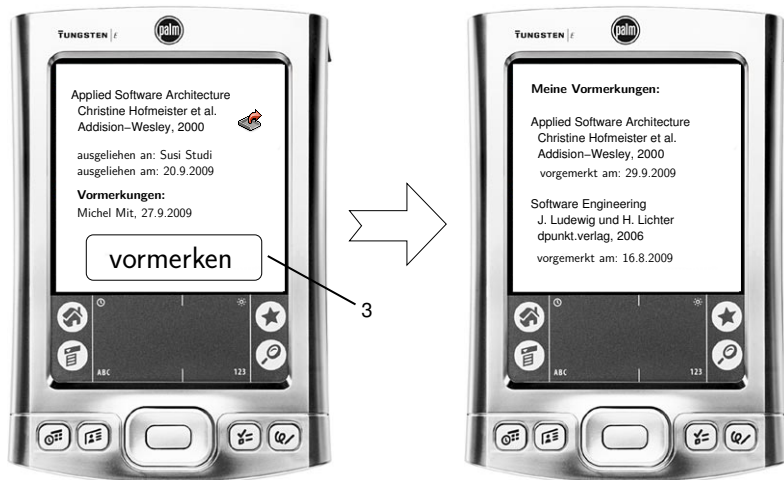


# Beispiel passives Storyboard



## Beispiel passives Storyboard

# Screen Vormerkung      Screen Vormerkungen



## Aktive Storyboards

### **Demonstration:**

Abspielen einer selbstablaufenden Präsentation des Systemverhaltens

### **Mittel:**

- Film, Diashow
- selbstablaufende Präsentation

### **Bemerkung:**

- + ermöglicht einfache automatisierte Darstellung von typischen Anwendungsszenarien
- + erfordert nicht unbedingt die Anwesenheit von Analytikern
- erlaubt keine Interaktion während der Präsentation

## Interaktive Storyboards

### **Demonstration:**

Prototyp ermöglicht dem Benutzer die frühzeitige Interaktion mit dem möglichen System; Funktionalität kann evtl. durch Analytiker „von Hand“ simuliert werden

### **Mittel:**

- ausführbares Programm, das Teile der Funktionalität realisiert

### **Bemerkung:**

- + ermöglicht Interaktion des Nutzers mit dem System
- + erlaubt größtmögliche Nähe zum realen System
- erfordert höheren Aufwand bei Prototyp-Erstellung

# Vor- und Nachteile des Einsatzes von Prototypen

- + erlauben frühzeitige Demonstration von Lösungsansätzen
- + erlauben frühzeitige Beteiligung der Benutzer
- + vermeiden das „Leere-Blatt-Syndrom“

# Vor- und Nachteile des Einsatzes von Prototypen

- + erlauben frühzeitige Demonstration von Lösungsansätzen
- + erlauben frühzeitige Beteiligung der Benutzer
- + vermeiden das „Leere-Blatt-Syndrom“
- + reduzieren Entwicklungsrisiken durch frühzeitige Diskussion mit Beteiligten
- + geeignete Werkzeuge ermöglichen die schnelle Erstellung von Prototypen



# Vor- und Nachteile des Einsatzes von Prototypen

- + erlauben frühzeitige Demonstration von Lösungsansätzen
- + erlauben frühzeitige Beteiligung der Benutzer
- + vermeiden das „Leere-Blatt-Syndrom“
- + reduzieren Entwicklungsrisiken durch frühzeitige Diskussion mit Beteiligten
- + geeignete Werkzeuge ermöglichen die schnelle Erstellung von Prototypen
- erfordern erhöhten Entwicklungsaufwand durch (zusätzliche) Prototyp-Entwicklung
- Gefahr, dass Wegwerf-Prototyp Teil des Produkts wird (z.B. aus Zeitdruck)



Wann wird welche Analysetechnik eingesetzt?

# Analysetechniken

	<b>Ist-Zustand</b>	<b>Soll-Zustand</b>	<b>Folgen</b>
Auswertung vorhandener Daten/Dokumente	+	-	-
Beobachtungen	+	o	-
Befragung			
- geschlossene Fragen	+	o	-
- offene Fragen	+	o	-
- hybride Fragen	+	o	-
Prototyping	-	+	+
partizipative Entwicklung	-	+	+

# Fragen



Wie halten wir die Anforderungen fest?

# Anforderungsspezifikation nach IEEE Std 610.12-1990

## Definition

**requirement:** condition or capability needed by a user to solve a problem or achieve an objective.

# Anforderungsspezifikation nach IEEE Std 610.12-1990

## Definition

**requirement:** condition or capability needed by a user to solve a problem or achieve an objective.

**specification:** document that specifies, in a complete, precise, verifiable manner, the requirements (, ...) of a system or component, and, often, the procedures for determining whether these provisions have been satisfied.

# Anforderungsspezifikation nach IEEE Std 610.12-1990

## Definition

**requirement:** condition or capability needed by a user to solve a problem or achieve an objective.

**specification:** document that specifies, in a complete, precise, verifiable manner, the requirements (, ...) of a system or component, and, often, the procedures for determining whether these provisions have been satisfied.

**software requirements specification (SRS):** documentation of the essential requirements (functions, performance, design constraints, and attributes) of the software and its external interfaces.

# Anforderungen

Anforderungen sind gleichbedeutend mit Minimalbedingungen hinsichtlich Funktion und Qualität.

⇒ Wir müssen also die Funktion und Qualität definieren.



# Anforderungen

Anforderungen sind gleichbedeutend mit Minimalbedingungen hinsichtlich Funktion und Qualität.

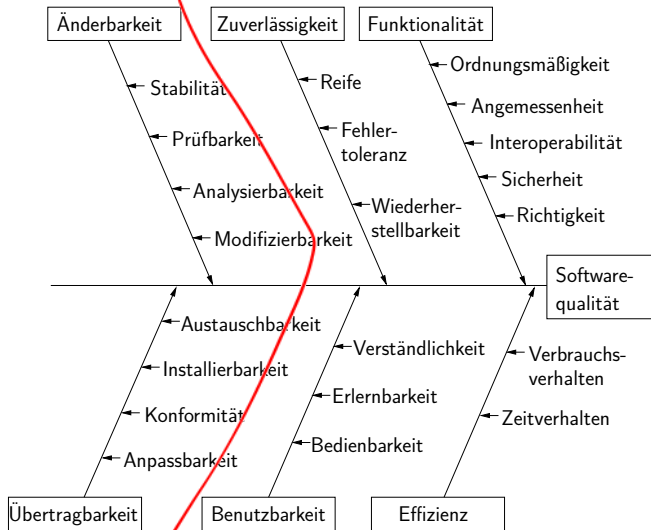
⇒ Wir müssen also die Funktion und Qualität definieren.

## Definition

**Funktion:** in der Zeit ablaufende Transformation

- von Eingabedaten
- in Ausgabedaten
- unter Verwendung von Ressourcen

# Produktqualitäten nach ISO/IEC-Standard 9126 (2001)



# Bedeutung einer Anforderungsspezifikation

## **Zweck**

Abstimmung mit  
Kunden

## **Folge von Mängeln**

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

## Bedeutung einer Anforderungsspezifikation

### **Zweck**

Abstimmung mit  
Kunden

Entwurf

### **Folge von Mängeln**

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.

## Bedeutung einer Anforderungsspezifikation

### **Zweck**

Abstimmung mit  
Kunden

Entwurf

Benutzerhandbuch

### **Folge von Mängeln**

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.

Basis für das Handbuch fehlt, es wird darum phänomenologisch verfasst.

## Bedeutung einer Anforderungsspezifikation

### **Zweck**

### **Folge von Mängeln**

Abstimmung mit  
Kunden

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

Entwurf

Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.

Benutzerhandbuch

Basis für das Handbuch fehlt, es wird darum phänomenologisch verfasst.

Testvorbereitung

systematischer Test ist unmöglich

## Bedeutung einer Anforderungsspezifikation

### **Zweck**

Abstimmung mit  
Kunden

Entwurf

Benutzerhandbuch

Testvorbereitung

Abnahme

### **Folge von Mängeln**

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.

Basis für das Handbuch fehlt, es wird darum phänomenologisch verfasst.

systematischer Test ist unmöglich

Korrektheit ist subjektiv, Streit ist unvermeidbar.

# Bedeutung einer Anforderungsspezifikation

## Zweck

Abstimmung mit  
Kunden

Entwurf

Benutzerhandbuch

Testvorbereitung

Abnahme

Wiederverwendung

## Folge von Mängeln

Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.

Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.

Basis für das Handbuch fehlt, es wird darum phänomenologisch verfasst.

systematischer Test ist unmöglich

Korrektheit ist subjektiv, Streit ist unvermeidbar.

nicht spezifizierte Systeme sind kaum durchschaubar, darum schwer wiederzuverwenden.



# Bedeutung einer Anforderungsspezifikation

<b>Zweck</b>	<b>Folge von Mängeln</b>
Abstimmung mit Kunden	Die Anforderungen bleiben ungeklärt, Wünsche des Kunden bleiben unberücksichtigt.
Entwurf	Entwerfer fehlt Vorgabe, darum mehr Kommunikation / eigene Vorstellung als Vorgabe.
Benutzerhandbuch	Basis für das Handbuch fehlt, es wird darum phänomenologisch verfasst.
Testvorbereitung	systematischer Test ist unmöglich
Abnahme	Korrektheit ist subjektiv, Streit ist unvermeidbar.
Wiederverwendung	nicht spezifizierte Systeme sind kaum durchschaubar, darum schwer wiederzuverwenden.
spätere Reimplementierung	Kompatibilität setzt voraus, dass man weiß, womit die neue Software kompatibel sein soll.

# Angestrebte Eigenschaften der Spezifikation

## inhaltlich

- (1) zutreffend (nicht „korrekt“!)
- (2) vollständig (relativ zu den Wünschen des Kunden)
- (3) widerspruchsfrei (oder konsistent, damit auch realisierbar)
- (4) neutral d.h. abstrakt (und damit offen für beliebigen Entwurf)

# Angestrebte Eigenschaften der Spezifikation

## inhaltlich

- (1) zutreffend (nicht „korrekt“!)
- (2) vollständig (relativ zu den Wünschen des Kunden)
- (3) widerspruchsfrei (oder konsistent, damit auch realisierbar)
- (4) neutral d.h. abstrakt (und damit offen für beliebigen Entwurf)

## in der Darstellung

- (5) leicht verständlich (für alle Zielgruppen!)
- (6) präzise (schließt Umgangssprache aus)

# Angestrebte Eigenschaften der Spezifikation

## inhaltlich

- (1) zutreffend (nicht „korrekt“!)
- (2) vollständig (relativ zu den Wünschen des Kunden)
- (3) widerspruchsfrei (oder konsistent, damit auch realisierbar)
- (4) neutral d.h. abstrakt (und damit offen für beliebigen Entwurf)

## in der Darstellung

- (5) leicht verständlich (für alle Zielgruppen!)
- (6) präzise (schließt Umgangssprache aus)

## in der Form


- (7) leicht erstellbar (was die Notationen und Modelle betrifft)
- (8) leicht verwaltbar (also auch zweckmäßig strukturiert)
- (9) objektivierbar (auch – nicht sinnvoll – „testbar“ genannt)

# Angestrebte Eigenschaften der Spezifikation

## inhaltlich

- (1) zutreffend (nicht „korrekt“!)
- (2) vollständig (relativ zu den Wünschen des Kunden)
- (3) widerspruchsfrei (oder konsistent, damit auch realisierbar)
- (4) neutral d.h. abstrakt (und damit offen für beliebigen Entwurf)

## in der Darstellung

- 
- (5) leicht verständlich (für alle Zielgruppen!)
  - (6) präzise (schließt Umgangssprache aus)

## in der Form

- (7) leicht erstellbar (was die Notationen und Modelle betrifft)
- (8) leicht verwaltbar (also auch zweckmäßig strukturiert)
- (9) objektivierbar (auch – nicht sinnvoll – „testbar“ genannt)

Diese Merkmale konkurrieren, d.h. die Erfüllung des einen erschwert oder verhindert die Erfüllung des anderen.

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben



# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren
- Ein Mengengerüst bilden

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren
- Ein Mengengerüst bilden
- Den Kunden (Benutzer) einbeziehen

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren
- Ein Mengengerüst bilden
- Den Kunden (Benutzer) einbeziehen
- Geeignete Sprachen und Werkzeuge verwenden

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren
- Ein Mengengerüst bilden
- Den Kunden (Benutzer) einbeziehen
- Geeignete Sprachen und Werkzeuge verwenden
- Die Spezifikation so früh wie möglich prüfen und dem Konfigurationsmanagement unterstellen

# Regeln für Analyse und Spezifikation

- Ein Begriffslexikon anlegen und entwickeln
- Von der Aufgabe ausgehen, nicht von ihrer Lösung
- Daten suchen, nicht Programmabläufe beschreiben
- Abstraktionsebene nicht in einer Darstellung wechseln
- Die Spezifikation nach Aspekten organisieren
- Ein Mengengerüst bilden
- Den Kunden (Benutzer) einbeziehen
- Geeignete Sprachen und Werkzeuge verwenden
- Die Spezifikation so früh wie möglich prüfen und dem Konfigurationsmanagement unterstellen
- Die Spezifikation intensiv verwenden

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 1. Einführung

- 1.1 Zweck
- 1.2 Rahmen
- 1.3 Definitionen, Akronyme und Abkürzungen
- 1.4 Referenzen
- 1.5 Übersicht über das Dokument

## 2. Allgemeine Beschreibung

- 2.1 Produktperspektive
- 2.2 Produktfunktionen
- 2.3 Charakteristika der Benutzer
- 2.4 Einschränkungen
- 2.5 Annahmen und Abhängigkeiten

## 3. Detaillierte Beschreibung



# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 1. Einführung

### 1.1 Zweck

- Zweck der Spezifikation
- adressierte Leser

### 1.2 Rahmen

- herzustellende Software mit Name
- was die Software tut und nicht tut
- Anwendung der Software mit Nutzen und Zielen

### 1.3 Definitionen, Akronyme und Abkürzungen

### 1.4 Referenzen

### 1.5 Übersicht über das Dokument

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

#### 2.1.4 Softwareschnittstellen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

#### 2.1.4 Softwareschnittstellen

#### 2.1.5 Kommunikationsschnittstellen (Netzwerkprotokolle etc.)

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

#### 2.1.4 Softwareschnittstellen

#### 2.1.5 Kommunikationsschnittstellen (Netzwerkprotokolle etc.)

#### 2.1.6 Speicherbeschränkungen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

#### 2.1.4 Softwareschnittstellen

#### 2.1.5 Kommunikationsschnittstellen (Netzwerkprotokolle etc.)

#### 2.1.6 Speicherbeschränkungen

#### 2.1.7 Betriebsoperationen (Operationsmodi, Dauer interaktiver und nichtinteraktiver Operationen, unterstützende Datenverarbeitungsfunktionen, Sicherungs- und Wiederherstellungsoperationen)



# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2. Allgemeine Beschreibung

### 2.1 Produktperspektive

Einbettung in ein Gesamtsystem

#### 2.1.1 Systemschnittstellen

#### 2.1.2 Benutzungsschnittstelle (logische Charakteristika. z.B. Monitorformat, Seiten- oder Fensterlayout, Inhalt von Berichten oder Menüs, Verfügbarkeit von Funktionstasten) sowie Regeln für die Schnittstelle

#### 2.1.3 Hardwareschnittstellen

#### 2.1.4 Softwareschnittstellen

#### 2.1.5 Kommunikationsschnittstellen (Netzwerkprotokolle etc.)

#### 2.1.6 Speicherbeschränkungen

#### 2.1.7 Betriebsoperationen (Operationsmodi, Dauer interaktiver und nichtinteraktiver Operationen, unterstützende Datenverarbeitungsfunktionen, Sicherungs- und Wiederherstellungsoperationen)

#### 2.1.8 Möglichkeiten der lokalen Anpassung

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2.2 Produktfunktionen

- Zusammenfassung der Funktionen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2.2 Produktfunktionen

- Zusammenfassung der Funktionen

## 2.3 Charakteristika der Benutzer

- Bildungsstand, Erfahrung, technische Kenntnisse

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2.2 Produktfunktionen

- Zusammenfassung der Funktionen

## 2.3 Charakteristika der Benutzer

- Bildungsstand, Erfahrung, technische Kenntnisse

## 2.4 Einschränkungen

Beispiele:

- Gesetzliche Rahmenbedingungen
- Hardwarebeschränkungen
- parallelisierte Ausführung
- erforderliche Zuverlässigkeit
- sicherheitskritische Aspekte
- Datenschutzaspekte

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 2.2 Produktfunktionen

- Zusammenfassung der Funktionen

## 2.3 Charakteristika der Benutzer

- Bildungsstand, Erfahrung, technische Kenntnisse

## 2.4 Einschränkungen

Beispiele:

- Gesetzliche Rahmenbedingungen
- Hardwarebeschränkungen
- parallelisierte Ausführung
- erforderliche Zuverlässigkeit
- sicherheitskritische Aspekte
- Datenschutzaspekte

## 2.5 Annahmen und Abhängigkeiten

- z.B. Betriebssystem ist auf Hardware verfügbar

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

- 3.1.1 Benutzungsschnittstelle

- 3.1.2 Hardwareschnittstelle

- 3.1.3 Softwareschnittstelle

- 3.1.4 Kommunikationsschnittstelle

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

- 3.1.1 Benutzungsschnittstelle

- 3.1.2 Hardwareschnittstelle

- 3.1.3 Softwareschnittstelle

- 3.1.4 Kommunikationsschnittstelle

### 3.2 Produktfunktionen

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

- 3.1.1 Benutzungsschnittstelle

- 3.1.2 Hardwareschnittstelle

- 3.1.3 Softwareschnittstelle

- 3.1.4 Kommunikationsschnittstelle

### 3.2 Produktfunktionen

### 3.3 Performanzanforderungen



# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

- 3.1.1 Benutzungsschnittstelle

- 3.1.2 Hardwareschnittstelle

- 3.1.3 Softwareschnittstelle

- 3.1.4 Kommunikationsschnittstelle

### 3.2 Produktfunktionen

### 3.3 Performanzanforderungen

### 3.4 Entwurfseinschränkungen

- z.B. Standards

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

3.1.1 Benutzungsschnittstelle

3.1.2 Hardwareschnittstelle

3.1.3 Softwareschnittstelle

3.1.4 Kommunikationsschnittstelle

### 3.2 Produktfunktionen

### 3.3 Performanzanforderungen

### 3.4 Entwurfseinschränkungen

z.B. Standards

### 3.5 Softwaresystemattribute

z.B. Zuverlässigkeit, Verfügbarkeit, Sicherheit, Wartbarkeit, Portabilität

# Inhalt der Anforderungsspezifikation nach IEEE Standard 830.1998

## 3. Detaillierte Beschreibung

### 3.1 Externe Schnittstellen

3.1.1 Benutzungsschnittstelle

3.1.2 Hardwareschnittstelle

3.1.3 Softwareschnittstelle

3.1.4 Kommunikationsschnittstelle

### 3.2 Produktfunktionen

### 3.3 Performanzanforderungen

### 3.4 Entwurfseinschränkungen

z.B. Standards

### 3.5 Softwaresystemattribute

z.B. Zuverlässigkeit, Verfügbarkeit, Sicherheit, Wartbarkeit, Portabilität

### 3.6 Andere Anforderungen

# Softwaresystemattribute

Softwaresystemattribute:

- oft als nicht-funktionale Anforderungen bezeichnet
- müssen objektivierbar sein

# Softwaresystemattribute

Softwaresystemattribute:

- oft als nicht-funktionale Anforderungen bezeichnet
- müssen objektivierbar sein

*Das System soll sicher sein.*

# Softwaresystemattribute

Softwaresystemattribute:

- oft als nicht-funktionale Anforderungen bezeichnet
- müssen objektivierbar sein

*Das System soll sicher sein.*

versus:

- PGP-Verschlüsselung wird verwendet
- Logging aller Aktionen
- Nachrichten dürfen nur über Verschlüsselungskomponente geschehen
- Indizierte Zugriffe auf Felder müssen zur Laufzeit geprüft werden

# Softwaresystemattribute

*Das System soll portierbar sein.*

# Softwaresystemattribute

*Das System soll portierbar sein.*

versus:

- Anteil der plattformabhängigen Komponenten  $< 2\%$
- Anteil der plattformabhängigen Codezeilen  $< 5\%$
- Verwendung einer portierbaren Hochsprache
- Einschränkung auf portierbare Sprachkonstrukte
- Verwendung eines verbreiteten Betriebssystems



# Präsentation der Anforderungen

Funktionale Anforderungen geordnet nach:

- Operationsmodus
  - ▶ z.B. Kontrollsysteme: Training, Normal, Notfall

# Präsentation der Anforderungen

Funktionale Anforderungen geordnet nach:

- Operationsmodus
  - ▶ z.B. Kontrollsysteme: Training, Normal, Notfall
- Benutzerklassen
  - ▶ z.B. Fahrzeugsteuerung: Fahrer, Fahrgäste, Wartungstechniker

# Präsentation der Anforderungen

Funktionale Anforderungen geordnet nach:

- Operationsmodus
  - ▶ z.B. Kontrollsysteme: Training, Normal, Notfall
- Benutzerklassen
  - ▶ z.B. Fahrzeugsteuerung: Fahrer, Fahrgäste, Wartungstechniker
- Objekte und Klassen
  - ▶ z.B. Patientenmonitorsystem: Patienten, Sensoren, Pflegepersonal, Räume, Ärztinnen, Medizin
  - ▶ jede Klasse wird beschrieben durch ihre Attribute und Methoden

# Präsentation der Anforderungen

Funktionale Anforderungen geordnet nach:

- Operationsmodus
  - ▶ z.B. Kontrollsysteme: Training, Normal, Notfall
- Benutzerklassen
  - ▶ z.B. Fahrzeugsteuerung: Fahrer, Fahrgäste, Wartungstechniker
- Objekte und Klassen
  - ▶ z.B. Patientenmonitorsystem: Patienten, Sensoren, Pflegepersonal, Räume, Ärztinnen, Medizin
  - ▶ jede Klasse wird beschrieben durch ihre Attribute und Methoden
- Features oder auch Anwendungsfälle (gewünschter nach außen sichtbarer Service)
  - ▶ z.B. Telefonsystem: Nahgespräch, Weiterleitung, Konferenzgespräch

# Präsentation der Anforderungen

Funktionale Anforderungen geordnet nach:

- Operationsmodus
  - ▶ z.B. Kontrollsysteme: Training, Normal, Notfall
- Benutzerklassen
  - ▶ z.B. Fahrzeugsteuerung: Fahrer, Fahrgäste, Wartungstechniker
- Objekte und Klassen
  - ▶ z.B. Patientenmonitorsystem: Patienten, Sensoren, Pflegepersonal, Räume, Ärztinnen, Medizin
  - ▶ jede Klasse wird beschrieben durch ihre Attribute und Methoden
- Features oder auch Anwendungsfälle (gewünschter nach außen sichtbarer Service)
  - ▶ z.B. Telefonsystem: Nahgespräch, Weiterleitung, Konferenzgespräch
- Stimuli (bei reaktiven Systemen)
  - ▶ z.B. Landesystem eines Flugzeugs: Energieverlust, Windwechsel, Schlingern